动态环境下基于人工势场引导的 RRT 路径规划算法 *

司徒华杰，雷海波，庄春刚

(上海交通大学 机械与动力工程学院，上海 200240)

摘 要：现有的大多数动态 RRT 路径规划算法不能使规划的路径远离障碍物，这有可能导致机器人没有足够的避障时间。针对这个问题，提出了一种利用人工势场引导快速扩展随机树向目标区域生长并远离障碍物的改进 RRT 算法 APFG-RRT(artificial potential field guided RRT)。然后，为了进一步加快算法的收敛速度，加速算法跳出局部极小值，引入了一种按自适应概率选择目标点作为采样点的策略。最后，针对动态环境采用全局规划结合局部重新规划的方法，以提高算法的实时性。仿真实验表明，相比于初始 RRT 和 Goal-bias RRT，APFG-RRT 的计算效率更高，内存需求更小，并且搜索到的路径能够有效地远离障碍物，提高了动态路径规划的成功率。

关键词：路径规划；RRT；人工势场；动态环境；局部重新规划

中图分类号：TP242 doi: 10.19734/j.issn.1001-3695.2020.02.0044

Artificial potential field based RRT algorithm for path planning in dynamic environment

Situ Huajie, Lei Haibo, Zhuang Chungang

(*School of Mechanical Engineering*, *Shanghai Jiao Tong University*, *Shanghai* 200240, *China*)

Abstract: Most of the existing dynamic RRT path planning algorithms cannot keep the planned path away from obstacles, which may cause the robot to have insufficient time to avoid obstacles. To solve this problem, this paper proposed an improved RRT, denoted as APFG-RRT, which utilized Artificial Potential Fields to guide the RRT grow to goal and away from obstacles. Then, in order to further increase the convergence rate and speed up the jump out of local minima, it introduced a strategy of selecting the goal as the random sample at an adaptive probability. Finally, it adopted global planning combined with local replanning to improve its real-time performance in dynamic environment. Simulation experiments show that APFG-RRT has higher computational efficiency and lower memory requirements compared with the initial RRT and Goal-bias RRT, and the given path can be effectively away from obstacles, which improves the success rate of dynamic path planning.

Key words: path planning; RRT; artificial potential fields; dynamic environments; local replanning

0 引言

路径规划是在给定的环境下寻找一条从初始位置到目标位置的无碰撞路径的过程，其在机器人系统[1]、医学[2]、计算机动画[3]、现代工业等众多领域应用广泛。根据对环境信息的掌握程度不同，路径规划可以分为全局路径规划和局部路径规划。局部路径规划算法有神经网络[4]、人工势场法[5]、遗传算法[6]等，其中人工势场法因其算法简单、实时性好、规划的路径平滑等优点而受到广泛关注。但人工势场法在规划路径时可能会陷入局部极小值，并且容易在目标点附近发生震荡，这大大限制了它的实际应用。全局路径规划的方法有蚁群算法[7]、RRT、可视图法[8]等，其中概率路标法[9]和 RRT 通过随机采样快速探索构型空间，有很高的计算效率，且具有概率完备性，即当规划的时间趋向于无穷时，算法找到一条可行路径的概率趋向于 1。

为了进一步提高 RRT 的收敛速度，许多研究人员在 RRT 的基础上进行改进。Kalisiak 等人提出 RRT-blossom[10]，用回归约束函数生成新节点，降低随机树前期重复探索同一区域的概率；Shkolnik 等人用超球体代替原来无体积的节点发展了 Ball Tree[11]，通过拒绝节点内的采样点使随机树专注于搜索还未探索的区域；Lavalle 等人提出了 Bidirectional RRT[12]，通过在初始位置和目标位置同时扩展随机树来加速探索速度；随后 Lavalle 还提出用目标点信息引导随机树偏向目标点生长的 Goal-bias RRT[13]；周芳等人引入贪婪生长的策略[14]，每次生长采用尽可能大的步长，从而快速覆盖更多区域。但上面提到的算法都没有利用环境信息，为了引进环境信息对 RRT 的引导，一些研究人员探索了将 RRT 与人工势场结合的方法。文献[15,16]通过切换两种算法将两者结合起来：当人工势场法规划路径陷入局部极小值时切换至 RRT 规划，而当跳出局部极小值时再切换回人工势场法；文献[17,18]提出将随机采样点向目标点方向偏移的改进 RRT*算法，偏移量与采样点和障碍物之间的最小距离相关，隐性地引入了斥力的大小；文献[19]将文献[18]的算法扩展到 B-RRT*中。由于这些算法是基于 RRT*进行改进，它们致力于搜索最优路径，实时性较差，不适用于实时性要求较高的动态路径规划。并且这些算法也不适合直接移植到只需规划出一条路径的动态 RRT 算法中，因为这些算法利用引力场和斥力的大小对采样点进行偏移，当随机树上节点较少时，采样点一般与待生长节点距离较远，采样点周围的势场不能表示待生长节点周围的环境信息，斥力对算法的引导作用较小，只有当随机树节点在搜索空间密集分布时，采样点周围的势场才近似等于待生长节点周围的势场。而且这些算法没有引入斥力的方向，这只能减缓随机树向障碍物的生长，而不能使随机树远离障碍物，这有可能使规划的路径贴近动态障碍物表面而有发生碰撞的危险。除此之外，这些算法的生长步长都是固定的，无法适应不同的局部环境。

收稿日期：2020-02-23；**修回日期：**2020-04-15 **基金项目：**国家自然科学基金资助项目(51775344)

作者简介：司徒华杰(1995-)，男，广东开平人，硕士研究生，主要研究方向为路径规划及机器视觉(hj.situ@sjtu.edu.cn)；雷海波(1996-)，男，畲族，浙江丽水人，硕士研究生，主要研究方向为机器视觉；庄春刚(1977-)，男，上海人，副研究员，博导，博士，主要研究方向为机器人学、机器视觉与控制.

为了克服上述问题，本文提出了用势场引导待生长节点生长的改进 RRT 动态路径规划算法——APFG-RRT，利用构造的引力势场和斥力势场使随机树向目标靠近的同时远离障碍物，并且在不同的局部环境下拥有自适应的生长步长。同时，本文提出了一种自适应概率采样的偏向生长策略，大大减少算法在局部极小值区域的重复计算，并针对动态环境使用了局部重新规划的策略。仿真实验表明，与原本的 RRT 和 Goal-bias RRT 相比，APFG-RRT 在包括狭窄和动态环境在内的各种环境下都有更高的计算效率和更少的节点数。

1　人工势场法

人工势场法是由 Khatib 提出的一种虚拟力法。它在机器人周围环境建立一个目标区域势能最小的虚拟势场，利用梯度下降引导机器人到达目标区域。具体为在目标区域构建一个引力势场 U_{att}，并在障碍物区域构建一个斥力势场 U_{rep}，使得机器人被吸引向目标区域而被障碍物区域排斥。引力势函数和引力函数通常为

$$U_{att}=\begin{cases}\frac{1}{2}K_a d^2(x,x_{goal}) & d(x,x_{goal})>d_g^*\\ K_a(d_g^* d(x,x_{goal})-\frac{1}{2}d_g^{*2}) & d(x,x_{goal})\le d_g^*\end{cases} \tag{1}$$

$$\boldsymbol{F}_{att}=\begin{cases}-K_a(x-x_{goal}) & d(x,x_{goal})>d_g^*\\ -K_a d_g^*\dfrac{x-x_{goal}}{d(x,x_{goal})} & d(x,x_{goal})\le d_g^*\end{cases} \tag{2}$$

其中，K_a 为引力常数，x 和 x_{goal} 分别为机器人控制点位置和目标位置，$d(\cdot)$ 表示两个位置之间的欧氏距离，为设定的距离阈值。当机器人控制点与目标点距离大于 d_g^* 时，引力势能与距离的二次方成正比，使机器人快速向目标点靠近；当控制点与目标点距离小于 d_g^* 时，引力势场以目标点为中心呈锥形，一定程度上缓解在目标点附近震荡的问题。

斥力势函数和斥力函数为

$$U_{rep}=\begin{cases}\frac{1}{2}K_r(\frac{1}{|\boldsymbol{d}_{min}|}-\frac{1}{d_o^*})^2 & |\boldsymbol{d}_{min}|\le d_o^*\\ 0 & |\boldsymbol{d}_{min}|>d_o^*\end{cases} \tag{3}$$

$$\boldsymbol{F}_{rep}=\begin{cases}K_r(\frac{1}{d_o^*}-\frac{1}{|\boldsymbol{d}_{min}|})\dfrac{\boldsymbol{d}_{min}}{|\boldsymbol{d}_{min}|^3} & |\boldsymbol{d}_{min}|\le d_o^*\\ 0 & |\boldsymbol{d}_{min}|>d_o^*\end{cases} \tag{4}$$

其中，K_r 为斥力常数，$\boldsymbol{d}_{min}$ 为由机器人控制点指向障碍物最近点的向量，d_o^* 为斥力势场的作用半径。

机器人的总势能和受到的合力为

$$U_{total}=U_{att}+U_{rep} \tag{5}$$

$$\boldsymbol{F}_{total}=\boldsymbol{F}_{att}+\boldsymbol{F}_{rep} \tag{6}$$

人工势场的小步长梯度下降算法如下所示，其中 x_{init} 为机器人的初始位置，ε 为设定的步长，函数 PotentialGradient 根据式(6)计算出所受合力。

算法 1　人工势场梯度下降搜索

1: $x_0 \leftarrow x_{init}$
2: while　$\nabla U_{total}\neq 0$:
3: 　$\boldsymbol{F}_{total}\leftarrow$ PotentialGradient(x_i)
4: 　$x_{i+1}\leftarrow x_i+\varepsilon\dfrac{\boldsymbol{F}_{total}}{|\boldsymbol{F}_{total}|}$
5: 　$i=i+1$

2　Goal-bias RRT

RRT 是一种基于增量采样的路径规划算法，其在机器人起始位置建立根节点，通过在机器人构型空间随机采样，引导随机树探索未覆盖的区域，最终找到一条通向目标点的路径。但是由于初始 RRT 采样的完全随机性，它在构型空间中的搜索漫无目的，导致效率很低。为此，许多启发式搜索的 RRT 算法被提出，Goal-bias RRT 是其中一种简单有效的算法。

初始的 RRT 首先将机器人初始位置 x_{init} 设为根节点，然后在机器人构型空间随机采样一个点 x_{rand}，选取随机树上离 x_{rand} 最近的节点 $x_{nearest}$ 作为待生长的点，由于 RRT 外部的节点对应的泰森多边形面积较大，所以外部节点得到生长的概率更大，这使得 RRT 拥有快速向外扩展的能力。接着由 $x_{nearest}$ 向 x_{rand} 方向以步长 ε 生长得到新的节点 x_{new}，判断 x_{new} 是否与障碍物发生碰撞，若碰撞则舍弃，若无碰撞则保留，不断重复以上步骤直到随机树生长到目标区域。而 Goal-bias-RRT 与初始 RRT 的采样点生成方式有所不同，Goal-bias RRT 按一定概率 P 选取目标点 x_{goal} 作为采样点，概率 $1-P$ 随机取点作为采样点，从而引导随机树向目标点生长。Goal-bias RRT 的伪代码如算法 2 所示。

算法 2　Goal-bias RRT

1: $T\leftarrow$ InitializeTree(x_{init})
2: for　$i=0$　to　K :
3: 　if　random$(0,1)<P$:
4: 　　$x_{rand}=x_{goal}$
5: 　else:
6: 　　$x_{rand}=$ Sample()
7: 　$x_{nearest}\leftarrow$ NearestNode(T,x_{rand})
8: 　$x_{new}\leftarrow$ Steer$(x_{nearest},x_{rand})$
9: 　if　CollisionFree(x_{new}) :
10: 　　T.add_node(x_{new})
11: 　　T.add_edge$(x_{nearest},x_{new})$
12: 　　if　$d(x_{new},x_{goal})<\varepsilon$:
13: 　　　break

3　基于人工势场引导的 RRT 路径规划算法

3.1　人工势场引导节点生长

Goal-bias RRT 加入了目标点信息，使 RRT 的搜索更有目的性，从而提高搜索效率。但是，它没有考虑到障碍物信息，而障碍物信息在大多场景都能够获取。为了加强算法对周围环境的感知，用障碍物信息进一步引导搜索，本文提出的 APFG-RRT 将人工势场融入到 RRT 中，根据目标点和周围障碍物信息构建人工势场，引导 RRT 进行搜索，使扩展随机树能够更有效地避开障碍物，从而快速搜索到到达目标的安全路径。

在 RRT 中，每次节点生长的增量仅由随机采样点 x_{rand} 相对 $x_{nearest}$ 的方向和步长 ε 决定，而在 APFG-RRT 中，生长的增量还由机器人在 $x_{nearest}$ 处受到的引力和斥力决定。APFG-RRT 新节点的生成公式如下：

$$x_{new}=x_{nearest}+\varepsilon\frac{x_{rand}-x_{nearest}}{d(x_{rand},x_{nearest})}+\phi\frac{\boldsymbol{F}_{total}}{|\boldsymbol{F}_{total}|} \tag{7}$$

其中，$\boldsymbol{F}_{total}$ 为机器人在 $x_{nearest}$ 处受到的引力和斥力的合力，ε 为随机分量因子，ϕ 为势场分量因子。需要注意的是，为了保持算法的概率完备性，使随机树能够到达构型空间中的任意无碰撞位姿，ϕ 和 ε 需要满足 $\phi<\varepsilon$ 。

与大多数 RRT 等步长的扩展方式不同，APFG-RRT 由随机分量和势场分量共同决定生长的距离，当节点离障碍物较近时，势场分量的方向主要由斥力决定，使得向障碍物区域的生长步长较小，有利于随机树通过狭窄区域；而当节点远离障碍物时，势场分量的方向主要由引力决定，使得向目标区域的生长步长较大，有利于快速接近目标。

APFG-RRT 的引力函数和斥力函数分别为

$$\boldsymbol{F}_{att}=F_{att}^*\frac{x_{goal}-x_{nearest}}{d(x_{goal},x_{nearest})} \tag{8}$$

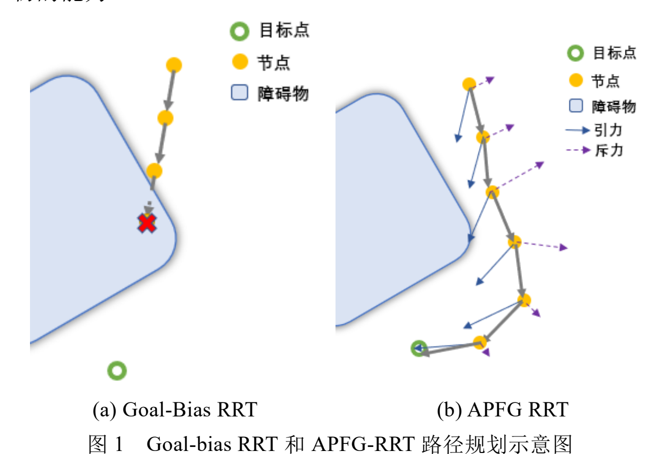
$$F_{rep}=\begin{cases}\dfrac{F_{rep}^{*}}{1+e^{(2|d_{\min}|/d_{rep}^{*}-1)k}}\dfrac{d_{\min}}{|d_{\min}|} & |d_{\min}|\le d_{rep}^{*}\\ 0 & |d_{\min}|>d_{rep}^{*}\end{cases}\tag{9}$$

其中，$\mathbf{d}_{min}=x_{nearest}-O_{nearest}$，$O_{nearest}$ 是障碍物上离 $x_{nearest}$ 最近的点。F_{att}^{*} 和 F_{rep}^{*} 分别为引力常数和斥力常数，k 是形状系数，d_{rep}^{*} 为斥力势场的作用半径。

人工势场法中引力大小之所以设置为式(2)是为了让机器人在远距离时快速靠近目标，而在接近目标时防止机器人过冲。而 RRT 一次规划出完整路径，在到达目标附近时直接将节点与目标相连即可得到路径，不会发生过冲，所以 APFG-RRT 中将引力的大小设置为常数 F_{att}^{*}。

APFG-RRT 中斥力函数为式(9)，当 $x_{nearest}$ 与障碍物的最小距离接近 0 时，斥力大小接近 F_{rep}^{*}；当 $x_{nearest}$ 与障碍物的最小距离接近 $d_{rep}^{*}/2$ 时，斥力大小接近 $F_{rep}^{*}/2$；当 $x_{nearest}$ 与障碍物的最小距离接近 d_{rep}^{*} 时，斥力大小接近 0。相比于式(4)，式(9)中参数的意义更加直观，斥力的分布更加合理。经过测试，斥力函数设为式(9)算法的性能更好。

假如将 Goal-bias RRT 和 APFG-RRT 中选取目标点作为采样点的概率都设置为 100%，则 Goal-bias RRT 和 APFG-RRT 在简单场景中的表现如图 1 所示。Goal-bias RRT 直接向目标生长，很容易在障碍物表面陷入局部极小值，需要依靠概率为 $1-P$ 的随机采样跳出局部极小值。而融合了人工势场的 APFG-RRT 由于加入了斥力机制，拥有了一定的绕开障碍物的能力。



(a) Goal-Bias RRT　　(b) APFG RRT

图 1　Goal-bias RRT 和 APFG-RRT 路径规划示意图

Fig. 1　Schematic diagrams of path planning with Goal-bias RRT and APFG-RRT

3.2　自适应概率采样

Goal-bias RRT 概率选取目标点作为采样点的策略不仅可以引导随机树向目标生长，还具有优先生长一条分支的特性，这有利于快速规划出一条路径。但是，Goal-bias RRT 选取目标点作为采样点的概率 P 是固定的，当随机树陷入局部极小值时，选取目标点作为采样点的扩展一般是无效的，这会浪费大量的计算资源。为此，本文提出了一种概率自适应的采样方法。当选取目标点作为采样点的扩展无效时，可认为随机树陷入局部极小值，将 P 设置为 0，使算法专注于通过随机生长跳出局部极小值，而随着随机生长的次数增加，随机树跳出局部极小值的可能性越来越大，所以相应地 P 也随着随机生长次数的增加而增加。随机树陷入局部极小值时 P 的值由下面公式获得：

$$P=P_{\max}(1-e^{-an/n^{*}})\tag{10}$$

其中，$P_{\max}$ 为随机树未陷入局部极小值时选取目标点作为采样点的概率，a 是形状系数，n^{*} 是设定的最大尝试次数，当随机生长次数接近 n^{*} 时，P 逐渐恢复成 $P_{\max}$。

3.3　静态环境下的 APFG-RRT 算法

APFG-RRT 的具体流程为：首先在机器人的初始位置建立随机树的根节点，将选择目标点作为采样点的概率 P 设为 $P_{\max}$，当随机数小于 P 时，选取目标点作为采样点 x_{rand}，而当随机数大于 P 时，在机器人构型空间中随机采样。然后选取随机树上离 x_{rand} 最近的节点 $x_{nearest}$ 作为待生长节点，计算出障碍物上离 $x_{nearest}$ 最近的点 $O_{nearest}$，根据式(7)计算出 x_{new}，判断 x_{new} 是否与障碍物碰撞，若碰撞且采样点为 x_{goal}，则认为随机树陷入局部极小值；若无碰撞则将 x_{new} 作为新节点添加到随机树中，并且如果采样点是 x_{goal}，则认为随机树没有陷入局部极小值或者已经跳出局部极小值，将 P 重新设为 $P_{\max}$。接着若随机树处于陷入局部极小值状态，则根据式(10)更新 P，不断重复上述步骤，直到随机树生长到目标点邻域或者迭代次数超过设定的最大值。静态环境下 APFG-RRT 的伪代码如下：

算法 3　静态环境下的 APFG-RRT 算法

```
1: T ← InitializeTree(x_init)
2: P = P_max, local_minima = False
3: for  i = 0  to  K :
4:     if  random(0,1) < P :
5:             x_rand = x_goal
6:     else:
7:             x_rand = Sample()
8:     x_nearest ← NearestNode(T, x_rand)
9:     O_nearest ← NearestObstacle(Obstacle, x_nearest, d*_rep)
10:    x_new ← Steer(x_nearest, x_rand, x_goal, O_nearest)
11:    if  CollisionFree(x_new) :
12:            T.add_node(x_new)
13:            T.add_edge(x_nearest, x_new)
14:            if  d(x_new, x_goal) < ε :
15:                   break
16:            if  x_rand == x_goal :
17:                   P = P_max
18:                   local_minima = False
19:     else:
20:            if  x_rand == x_goal :
21:                   n = 0
22:                   local_minima = True
23:     if  local_minima :
24:            P ← UpdateProbability(n)
25:            n = n + 1
```

3.4　动态环境下的 APFG-RRT 算法

动态环境下路径规划的难点在于障碍物可能会运动到之前规划好的路径上，需要算法在极短时间内重新规划出一条无碰撞路径，因而对算法的实时性要求很高。另一方面，考虑到需要留给机器人一定的避障时间和空间，规划的路径还应远离障碍物。传统的膨胀障碍物的方法虽然能使规划路径与障碍物保持一定的安全距离，但是对于不同场景需要调整膨胀参数，比如在狭窄的环境需要选择较小的膨胀程度，否则将会使随机树的扩展更加艰难。而 APFG-RRT 由于加入了势场对随机树生长的引导机制，具有更快的收敛速度，且生成的路径能够相对远离障碍物，在动态环境下有较好的表现。

为了进一步提高算法在动态环境下的实时性，本文参考 *RRT*FND* [20]中的方法，使用了局部重新规划的策略——只对在当前位置一定范围内未执行的“危险”路径进行重新规划，从而减少计算量。具体为先搜索出在当前位置半径为 r 范围内的将要执行的连续路径 p_{near}，将离障碍物最小距离在 d^{*} 以内的节点认为是“危险”节点，用 APFG-RRT 对 p_{near} 上包含所有“危险”节点的最短路径段进行重新规划和连接。算法的伪代码如算法 4 所示。

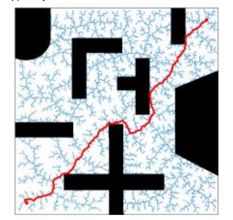
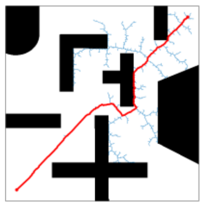
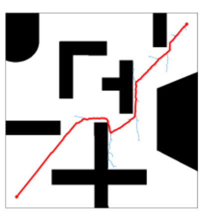
算法 4　动态环境下的 APFG-RRT 算法

```
1: Obstacle ← UpdateObstacles()
2: p ← APFG-RRT(x_init, x_goal, Obstacle)
3: x_current ← x_init
4: while x_current ≠ x_goal :
5:     p_near ← SelectNearPath(p)
6:     x_start, x_end ← DangerousPath(p_near)
7:     p_rest ← DeletePath(p, x_start, x_end)
8:     p_changed ← APFG-RRT(x_start, x_end, Obstacle)
9:     P ← AddPath(p_rest, p_changed)
10:    Obstacle ← UpdateObstacles()
11:    x_current ← GetCurrentState()
```

4　实验结果与分析

为了测试 APFG-RRT 的收敛速度，本文对 APFG-RRT 在多种环境下进行了仿真实验，并与初始的 RRT 和 Goal-bias RRT 进行比较。三种算法中相应的参数保持相同。考虑到 RRT 自身的随机性，在对应一般环境和狭窄环境的地图上重复进行 100 次由起点到目标点的路径规划，取平均值作为测试结果。

图 2 和 3 分别展示了初始 RRT、Goal-bias RRT 和 APFG-RRT 在一般场景和狭窄场景中的表现，表 1 和 2 则列出了相应的实验数据。从表中可以看出，Goal-bias RRT 和 APFG-RRT 的表现明显优于初始的 RRT，而 APFG-RRT 在各项指标上均优于 Goal-bias RRT。在地图 1(一般场景)中 APFG-RRT 路径规划的平均时间为 Goal-bias RRT 的 49.3%，平均节点数为 Goal-bias RRT 的 41.5%；在地图 2(狭窄场景)中 APFG-RRT 路径规划的平均时间为 Goal-bias RRT 的 47.2%，平均节点数为 Goal-bias RRT 的 42.9%。这是因为人工势场的引入虽然增加了算法每次迭代的计算量，但使 APFG-RRT 拥有绕开障碍物的能力，减少了与障碍物发生碰撞的无效节点数，而自适应概率采样策略使因陷入局部极小值而重复计算的无效节点数进一步减少。从表中数据可计算出 Goal-bias RRT 在地图 1、2 中的平均无效节点数分别为 5189 和 5555，而 APFG-RRT 在地图 1、2 中的平均无效节点数仅为 164 和 12，这大大减少了算法无效迭代所浪费的时间。另一方面，随机树上已有的节点数越多，算法每次迭代时搜索随机树上离采样点最近的节点消耗的时间越多。Goal-bias RRT 按概率 1-P 随机采样，而 APFG-RRT 在概率 1-P 时由势场分量和随机分量共同决定生长方向，这使得 APFG-RRT 的扩展更有目的性，所以 APFG-RRT 搜索到路径时随机树上的节点数更少，这减少了平均每次迭代消耗的时间。对于狭窄场景，Goal-bias RRT 因偏向目标生长的特性而容易陷入贴近障碍物表面、取目标点为采样点的扩展失效而需要依赖概率为 1-P 的随机采样逃脱的循环中，而在贴近狭窄通道表面时随机树的生长方向受到限制，支持有效生长的采样空间较小；而 APFG-RRT 因狭窄通道中间的势能最小，容易使随机树沿着通道中间生长，使得随机树的生长方向受到的限制较小，有效生长的概率更高，而且当生长的随机分量朝向通道表面时，势场分量远离障碍物，使得生长步长减小，有利于进一步提高有效生长的概率。

(a)Initial RRT　(b) Goal-bias RRT　(c) APFG-RRT

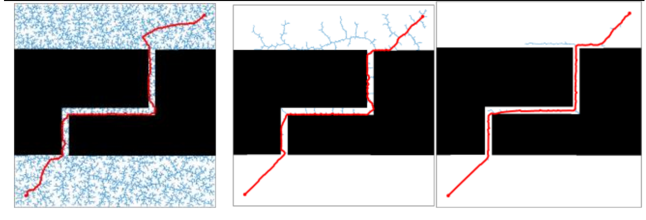
图 2　三种算法在地图 1(一般场景)中的表现

Fig. 2　Performance of the three algorithms in Map 1(general scene)

表 1　三种算法在一般场景中的实验结果

Tab. 1　Experimental results of the three algorithms in general scene

算法	RRT	Goal-bias RRT	APFG-RRT
平均规划时间/s	33.81	0.414	0.204
平均迭代次数	37281	7617	1172
平均节点数	26895	2428	1008



(a) Initial RRT　(b) Goal-bias RRT　(c) APFG-RRT

图 3　三种算法在地图 2(狭窄环境)中的表现

Fig. 3　Performance of the three algorithms in Map 2(narrow environment)

表 2　三种算法在狭窄环境中的实验结果

Tab. 2　Experimental results of the three algorithms in narrow environment

算法	RRT	Goal-bias RRT	APFG-RRT
平均规划时间/s	39.869	0.303	0.143
平均迭代次数	54274	7284	754
平均节点数	27337	1729	742

图 4 展示了 APFG-RRT 在动态场景中使用局部重新规划策略后的表现。图中灰色的是运动障碍物，它们在固定的路线上往复运动，箭头为它们在所在帧中的运动方向，黄色粗实线为机器人已经执行的路径，红色点线为上一帧规划路径中被修正的部分，蓝色细实线为机器人在所在帧中重新规划后的路径。从图 4 可以看出：加入势场引导生长的机制后 APFG-RRT 规划的路径相对远离障碍物，留给了机器人一定的避障空间。值得注意的是，局部重新规划能够在一定程度上优化原来的路径，比如在图 4 的第 3 张图中重新规划的路径比原来的路径更短、更平滑，并且使原来由于陷入局部极小值而靠近障碍物的路径远离障碍物，这是因为势场的特性使得 APFG-RRT 在局部简单环境中的规划具有良好的表现。

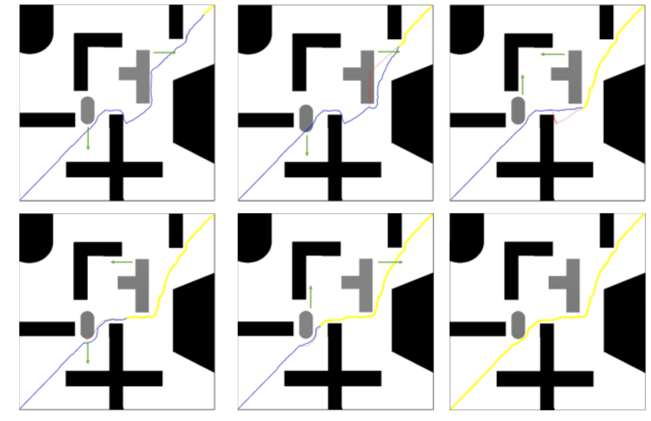


图 4　APFG-RRT 在动态环境中的表现

Fig. 4　Performance of APFG-RRT in dynamic environment

在实验中，Goal-bias RRT 在动态场景中的路径规划采用对每帧所有路径都进行重新规划的策略。两种算法在图 4 场景中规划 100 次，并取平均值的规划结果如表 3 所示。

表 3　Goal-bias RRT 和 APFG-RRT 在动态环境中的实验结果

Tab. 3　Experimental results of Goal-bias RRT and APFG-RRT in dynamic environment

算法	Goal-bias RRT	APFG-RRT
每次重新规划的平均时间/s	0.421	0.011
规划成功率	0	100%

由于 Goal-bias RRT 具有偏向目标点生长的特性，其搜索到的路径容易在局部极小值区域贴合障碍物，可能导致机器人与障碍物发生碰撞。如图 5 所示，Goal-bias RRT 在当前帧中重新规划的路径虽然是无碰撞的，但是贴合着障碍物，且在障碍物运动方向上，在下一帧时刻障碍物就会与机器人发生碰撞，这就是导致 Goal-bias RRT 在实验的动态场景中规划成功率为 0 的原因。而对于其他大多数的 RRT 算法，由于没有远离障碍物的机制，规划的路径也有可能会贴合着障碍物，比如图 2(a)中初始 RRT 规划的路径。

由于 Goal-bias RRT 规划成功率为 0，表 3 中 Goal-bias RRT 每次重新规划的平均时间是统计其中成功重新规划的部分得到。从实验结果可以看出，采用局部重新规划策略的 APFG-RRT 在测试的动态场景中规划成功率为 100%，且每次重新规划的平均时间远少于 Goal-bias RRT。这是因为相比于在每一帧时刻都对全局路径重新进行规划的算法，局部重新规划使得算法在每一帧时刻只需重新规划其中一小段路径，当将要执行的路径安全时甚至不需要重新进行规划，这大大减少了计算量，而引入局部重新规划策略而额外增加的计算量相比之下很小。

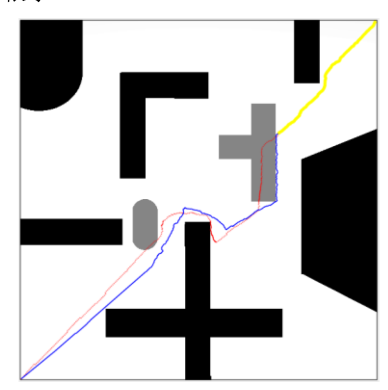


图 5 Goal-bias RRT 应用于动态环境时容易在局部极小值区域与障碍物碰撞

Fig. 5 When Goal-bias RRT is applied to dynamic environment, it is easy to collide with obstacles in the local minimum area

5 结束语

本文针对现有的大多数 RRT 动态路径规划算法不能使规划的路径与障碍物保持安全距离的问题，提出了一种用人工势场引导 RRT 生长的启发式算法，增强了算法对周围环境的感知能力，并结合自适应概率采样和局部重新规划策略，使 RRT 的搜索效率进一步提高。仿真结果表明，本文提出的改进 RRT 算法在静态和动态场景下都有较为优越的性能。

参考文献：

[1] Tzafestas, Spyros G. Mobile Robot Control and Navigation: A Global Overview [J]. Journal of Intelligent & Robotic Systems, 2018, 91 (1): 35-58.

[2] Kong Xiangzhan, Duan Xingguang, Wang Yonggui. An integrated system for planning, navigation and robotic assistance for mandible reconstruction surgery [J]. Intelligent Service Robotics, 2016, 9 (2): 113-121.

[3] Elbanhawi M, Simic M. Sampling-Based Robot Motion Planning: A Review [J]. IEEE Access, 2014, 2 (2): 56-77.

[4] Zhang Yinyan, Li Shuai, Guo Hongliang. A type of biased consensus-based distributed neural network for path planning [J]. Nonlinear Dynamics, 2017, 89 (3): 1803-1815.

[5] Bounini F, Gingras D, Pollart H, *et al.* Modified artificial potential field method for online path planning applications [C]// Proc of IEEE Intelligent Vehicles Symposium (IV) . Piscataway, NJ: IEEE Press, 2017: 180-185.

[6] Li Jinghua, Huang Yibin, Xu Zhao, *et al.* Path planning of UAV based on hierarchical genetic algorithm with optimized search region [C]// Proc of IEEE International Conference on Control & Automation. Piscataway, NJ: IEEE Press, 2017: 1033-1038.

[7] Liu Yang, Ma Jianwei, Zang Shaofei, *et al.* Dynamic Path Planning of Mobile Robot Based on Improved Ant Colony Optimization Algorithm [C]// Proc of the 8th International Conference on Networks, Communication and Computing. New York: ACM Press, 2019: 248-252.

[8] Latip N B A, Omar R, Debnath S K. Optimal Path Planning using Equilateral Spaces Oriented Visibility Graph Method [J]. International Journal of Electrical & Computer Engineering, 2017, 7 (6): 3046-3051.

[9] Kumar N, Vámossy Z, Szabó-Resch Z M. Robot path pursuit using probabilistic roadmap [C]// Proc of IEEE International Symposium on Computational Intelligence & Informatics. Piscataway, NJ: IEEE Press, 2016: 139-144.

[10] Kalisiak M, Panne M V D. RRT-blossom: RRT with a local flood-fill behavior [C]// Proc of IEEE International Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2006: 1237-1242.

[11] Perez A, Karaman S, Shkolnik A, *et al.* Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ: IEEE Press, 2011: 4307-4313.

[12] Lavalle S M, Kuffner J J. Randomized kinodynamic planning [C]// Proc of IEEE International Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 1999: 473-479.

[13] Lavalle S M. Planning Algorithms [M]. Cambridge: Cambridge University Press, 2006.

[14] 周芳，朱齐丹，赵国良. 基于改进快速搜索随机树法的机械手路径优化 [J]. 机械工程学报, 2011, 47 (11): 30-35. (Zhou Fang, Zhu Qidan, Zhao Guoliang. Path Optimization of Manipulator Based on the Improved Rapidly-exploring Random Tree Algorithm [J]. Journal of Mechanical Engineering, 2011, 47 (11): 30-35.)

[15] 何兆楚，何元烈，曾碧. RRT 与人工势场法结合的机械臂避障规划 [J]. 工业工程, 2017, 20 (2): 56-63. (He Zhaochu, He Yuanlie, Zeng Bi. Obstacle Avoidence Path Planning for Robot Arm Based on Mixed Algorithm of Artificial Potential Field Method and RRT [J]. Industrial Engineering Journal, 2017, 20 (2): 56-63.)

[16] 徐晓慧，张金龙. 改进人工势场与 TAS-RRT 融合优化算法 [J]. 电子技术应用, 2018, 44 (10): 88-92. (Xu Xiaohui, Zhang Jinlong. Hybrid optimization algorithm of improved artificial potential field and TAS-RRT [J]. Application of Electronic Technique, 2018, 44 (10): 88-92.)

[17] Qureshi A H, Mumtaz S, Iqbal K F, *et al.* Adaptive Potential guided directional-RRT [C]// Proc of IEEE International Conference on Robotics and Biomimetics. Piscataway, NJ: IEEE Press, 2013: 1887-1892.

[18] Qureshi A H, Ayaz Y. Potential functions based sampling heuristic for optimal path planning [J]. Autonomous Robots, 2016, 40 (6): 1079-1093.

[19] Zaid T, Qureshi A H, Yasar A, *et al.* Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments [J]. Robotics and Autonomous Systems, 2018, 108: 13-27.

[20] Adiyatov O, Varol H A. A novel RRT*-based algorithm for motion planning in Dynamic environments [C]// Proc of IEEE International Conference on Mechatronics and Automation. Piscataway, NJ: IEEE Press, 2017: 1416-1421.